WORKMAN NYDEGGER
A PROFESSIONAL CORPORATION
ATTORNEYS AT LAW
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84111
TELEPHONE (801) 533-9800
FAX (801) 328-1707

TELECOPIER COVER SHEET

November 9, 2009

Total Number of Pages
      (including cover letter):   10 pages

Please deliver the transmitted facsimile pages to:

Examiner Cheneca Smith

United States Patent & Trademark Office

Business Phone	(571) 270-1651
Telecopier Phone:	(571) 270-2651
From:	Chantell Johnston for Brian Tucker
Comments:	Please see the attached draft amendment to be disussed in the In-Person Interview on Wednesday, November 18, 2009 at 1:30 PM
Serial No.	10/808,905
Docket No.	13768.1256

DO NOT ENTER ON THE RECORD

****

We are transmitting from a Sharp FO-750 or Sharp FO-6100 facsimile machine. If you do not receive all the pages or they are unreadable, please contact me as soon as possible at (801) 533-9800.
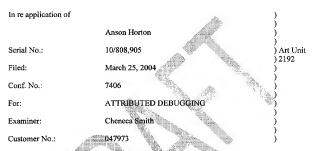
JML0000000637V001

BRIAN TUCKER
OFFICE DIRECT: 801-322-8427
EMAIL: btucker@WNLAW.com

ELECTRONICALLY FILED

PATENT APPLICATION
Docket No. 13768.1256

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of		)
		)
	Anson Horton	)
		)
Serial No.:	10/808,905	) Art Unit
		) 2192
Filed:	March 25, 2004	)
		)
Conf. No.:	7406	)
		)
For:	ATTRIBUTED DEBUGGING	)
		)
Examiner:	Cheneca Smith	)
		)
Customer No.:	047973	)

AMENDMENT "F" AND RESPONSE AFTER NON-FINAL

ELECTRONICALLY FILED
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In response to the Office action mailed October 2, 2009 (paper no. 20090928), please amend the above-identified application as follows:

Amendments to the Claims are reflected in the listing of claims which begins on page 2 of this paper;

Remarks/Arguments begin on page 7 of this paper.

AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application:

Listing of Claims:

1.   (Currently Amended) A computing system to facilitate attributed debugging for debugging a computer software application, the computing system comprising:

a processor; and

a computer readable storage medium storing the following components:

a debugger that facilitates debugging of a computer software application;

a debuggee that includes one or more attributes associated with the computer software application, which attributes are employed by the debugger to facilitate debugging of the software application, wherein the attributes annotate one or more of a type, field, method, or property of the software application, and wherein each of the one or more attributes contains a field, property, or method that is evaluated during debugging to customize the display of the corresponding annotated type, field, method, or property; and,

an expression evaluator that evaluates the field, property, or method of each of the one or more attributes ~~associated with the computer software application according to an attribute definition~~, and presents debug information ~~associated with the computer software application in accordance with the attribute definition~~representing the corresponding type, field, method, or property based on the evaluation of the field, property, or method within the associated attribute.~~, wherein the evaluation of at least one of the one or more attributes determines that a value to be shown for a field or property of the software application is the result of an evaluation of an expression contained in the code of the software application, which value is different from the value that would be shown for the field or property when the corresponding attribute is removed, and wherein the attribute definition declaratively indicates the developer-customizable format in which the debug information is displayed in a developer-customizable data window.~~

2.   (Previously Presented) The system of claim 1, wherein the expression evaluator evaluates an expression associated with a particular programming language.

3.   (Previously Presented) The system of claim 2, wherein the programming language is at least one of C#, J# or Visual Basic.Net.

4.   (Previously Presented) The system of claim 2, wherein the computer readable storage medium stores a plurality of expression evaluators, each expression evaluator associated with a particular programming language.

5.   (Currently Amended) The system of claim 1, wherein the field, property, or method of one or the one or more attributes is an overridden implementation of a ToString method, and wherein the expression evaluator invokes an the overridden implementation of a the ToString method to facilitate presentationpresent of the debug information for the corresponding annotated type, field, method, or property.

6.   (Currently Amended) The system of claim 5, wherein the expression evaluator employs a result of the overridden ToString method as a value to be displayed for an object, field, property, or combinations thereof with the debug information.

7.   (Currently Amended) The system of claim 1, wherein at least one of the one or more attributes determinesspecifies at least one of how and whether a type or memberthe corresponding annotated type, field, method, or property is displayed during debugging.

8.   (Currently Amended) The system of claim 7, wherein at least one of the one or more attributes employs an enumeration.

9.   (Currently Amended) The system of claim 8, wherein one enumeration value associated with an indication specifies that the type or membercorresponding annotated type, field, method, or property should not be displayed to the developer.

Page 3 of 9

Application No. 10/808,905
Amendment "F" dated eFormDatas
Reply to Non-Final Office Action mailed October 2, 2009

10.   (Previously Presented) The system of claim 8, wherein one enumeration value associated with an indication specifies that if a type is hierarchical, it should be expanded by default.

11.   (Previously Presented) The system of claim 8, wherein one enumeration value associated with an indication specifies that a type should not be expanded by default.

12.   (Previously Presented) The system of claim 8, wherein one enumeration value associated with an indication specifies that a target element itself should not be shown, but should instead be automatically expanded to have its member(s) displayed.

13.   (Currently Amended) The system of claim 1, wherein at least one of the one or more attributes includes a string in conjunction with the field, property, or method which string is displayed in conjunction with the results of the evaluated field, property, or method.specifies what is displayed for a class and/or field.

14.   (Currently Amended) The system of claim 13, wherein an argument to the attribute comprises athe string that is displayed in a value column for an instance of a corresponding the class and/or field.

15.   (Canceled)

16.   (Previously Presented) The system of claim 1, further comprising an attribute cache directory that stores an attribute associated with the computer software application, the expression evaluator employing the stored attribute to present debug information.

Page 4 of 9

17.   (Currently Amended) A computer-implemented method for facilitating debugging a computer software application using attributed debugging, the method comprising:

determining ~~whether~~ that a ~~process in~~ type, field, method, or property of a computer software application has an attribute attached thereto, wherein an attributes annotates one or more of a type, field, method, or property of the software application, and wherein each attribute contains a field, property, or method that is evaluated during debugging to customize the display of the corresponding annotated type, field, method, or property;

~~if the process has an attribute attached thereto,~~ evaluating the field, property, or method of the attribute; and ~~according to a definition of the attribute to display debug information, wherein the evaluation of the definition of the attribute determines that a value to be shown for a field or property of the software application is the result of an evaluation of an expression contained in the code of the software application, which value is different from the value that would be shown for the field or property when the corresponding attribute is removed; and~~

displaying ~~the~~ debug information representing the corresponding type, field, method, or property based on the evaluation of the field, property, or method within the associated attribute. ~~in a developer-customizable format according to declarative indication included in the attribute.~~

18.   (Currently Amended) The method of 17, further comprising at least one of the following:

determining ~~whether~~ that a ToString method has been overridden; and,

invoking the overridden ToString method to generate the debug information that is displayed ~~facilitate debugging, if the ToString method has been overridden.~~

19.   (Previously Presented) A computer readable storage medium having stored thereon computer executable instructions for carrying out the method of claim 17.

20.   (Canceled)

21.   (Currently Amended) A computer readable storage medium storing the computer executable components of claim 1.~~an attributed debugging system, the attributed debugging system comprising:~~

~~a debugger component that allows the manipulation of view data in the debugger; and,~~

~~an expression evaluator component that employs an attribute definition to evaluate an attribute associated with the computer software application to present debug information associated with the computer software application to a developer, wherein the employment of the attribute definition determines that a value to be shown for a field or property of the software application is the result of an evaluation of an expression contained in the code of the software application, which value is different from the value that would be shown for the field or property when the corresponding attribute is removed, and wherein the debug information is displayed in a developer-customizable format in accordance with the attribute definition and in a developer-customizable data window.~~

22.   (Canceled)

Application No. 10/808,905
Amendment "F" dated eFormData
Reply to Non-Final Office Action mailed October 2, 2009

REMARKS

The Office Action mailed October 2, 2009, considered and rejected claims 1-19, 21 and 22. Claims 1, 5-7, 13-17-21 were rejected under 35 U.S.C. § 103(a) as being unpatentable over *Bates et al.* (U.S. Publ. No. 2003/0221185) in view of *Bates et al.* (U.S. Patent No. 7,251,808) hereinafter *Bates_2.* Claim 22 was rejected under 35 U.S.C. § 103(a) as being unpatentable over *Bates et al.* in view of *Bates_2* and *Kobayashi et al.* (U.S. Patent No. 6,633,888). Claims 2-4 and 8-12 were rejected under 35 U.S.C. § 103(a) as being unpatentable over *Bates et al.* in view of *Bates_2,* as applied to claims 1 and 7 above, and further in view of *Dandoy* (U.S. Publ. No. 2004/0230954). [1]

By this response, claims 1, 5-9, 13-14, 17, 18, and 21 are amended, while claim 15 is canceled. Claims 1-14, 16-19, and 21 remain pending of which claims 1 and 17 are independent.

The present invention is directed to embodiments for using attributes to annotate source code in a manner that specifies how debugging information for the annotated source code is to be displayed during debugging. The claim amendments have been made to better clarify what attributes are as well as what they contain. For example, the independent claims recite that an attribute annotates a type, field, method, or property. Each attribute contains either a field, property, or method that is evaluated to determine what is displayed for the corresponding annotated type, field, method, or property. An example of this kind of attribute is shown on line 19 of page 8 of the specification.

The cited art fails to teach or suggest each aspect of the claims. For example, neither of the Bates references discloses the use of attributes. The closest thing disclosed is the display of comments that are proximate to the line of code that defines a variable as is shown in figure 7 of Bates-1. This figure shows that when the variable "stuff.y" is hovered over, its associated comment "stuff is important to output" is displayed. *See* ¶ 67. However, comments as used in Bates-1 are simply text that is meaningless when compiled and do not affect the source code in any way. In contrast, the present invention involves the use of attributes which are accessed and evaluated with the source code. In this manner, the field, property, or method inside the attributes can be evaluated and affect what is displayed for a corresponding type, field, method,

---

[1] Although the prior art status of the cited art is not being challenged at this time, Applicant reserves the right to challenge the prior art status of the cited art at any appropriate time, should it arise. Accordingly, any arguments and amendments made herein should not be construed as acquiescing to any prior art status of the cited art.

Application No. 10/808,905
Amendment "F" dated eFormDates
Reply to Non-Final Office Action mailed October 2, 2009

or property. In short, the Bates references fail to teach or suggest: "a debuggee that includes one or more attributes associated with the computer software application, which attributes are employed by the debugger to facilitate debugging of the software application, wherein the attributes annotate one or more of a type, field, method, or property of the software application, and wherein each of the one or more attributes contains a field, property, or method that is evaluated during debugging to customize the display of the corresponding annotated type, field, method, or property; and, an expression evaluator that evaluates the field, property, or method of each of the one or more attributes, and presents debug information representing the corresponding type, field, method, or property based on the evaluation of the field, property, or method within the associated attribute," as claimed in combination with the other limitations of independent claim 1.

In view of the foregoing, Applicant respectfully submits that the other rejections to the claims are now moot and do not, therefore, need to be addressed individually at this time. It will be appreciated, however, that this should not be construed as Applicant acquiescing to any of the purported teachings or assertions made in the last action regarding the cited art or the pending application, including any official notice. Instead, Applicant reserves the right to challenge any of the purported teachings or assertions made in the last action at any appropriate time in the future, should the need arise. Furthermore, to the extent that the Examiner has relied on any Official Notice, explicitly or implicitly, Applicant specifically requests that the Examiner provide references supporting the teachings officially noticed, as well as the required motivation or suggestion to combine the relied upon notice with the other art of record.

In the event that the Examiner finds remaining impediment to a prompt allowance of this application that may be clarified through a telephone interview, the Examiner is requested to contact the undersigned attorney at (801) 533-9800.

Dated this ___ day of _____, 2009.

Respectfully submitted,

RICK D. NYDEGGER
Registration No. 28,651

Page 8 of 9

Application No. 10/808,905
Amendment "F" dated ‹FormData›
Reply to Non-Final Office Action mailed October 2, 2009

BRIAN D. TUCKER
Registration No. 61,550
Attorneys for Applicant
Customer No. 47973

RDN:BDT:cj
2588868_1

Page 9 of 9